



Web

Results 11 - 20 of about 299,000 for **template return sizeof**. (0.18 seconds)

[idxl_layout.C Source File](#)

... **sizeof**(int); 00023 case IDXL_INDEX_1 : **return sizeof**(int); 00024 ... default: break;
00039 } 00040 **return** "(unknown IDXL ... 00047 **template**<class T> 00048 inline void ...
charm.cs.uiuc.edu/doxygen/charm/idxl_layout_8C-source.html - 39k - [Cached](#) - [Similar pages](#)

[TAU Portable Profiling Package ** * <a href="http://www.acl.lanl. ...](#)

... **template** <class T> int func_tmpl(Tx) { TAU_TYPE_STRING(buf, "int (" + CT(x) + ")"
TAU_PROFILE("func_tmpl", buf, TAU_PAWS3 | TAU_USER); **return sizeof**(x ...
www.cs.uoregon.edu/research/paracomp/tau/tautools/docs/example.cpp - 9k - [Cached](#) - [Similar pages](#)

[Florian Schintke - Re: The future C++ **template** model in gcc](#)

... In general this is not possible, consider (on an ILP=32 machine) > **template** <typename
T> int Foo () {**return sizeof** (T);} > Foo<char> () // **return** 1 > Foo<int ...
gcc.gnu.org/ml/gcc/2001-07/msg01913.html - 7k - [Cached](#) - [Similar pages](#)

[Extreme Cpp Form Test](#)

... is not the correct size: " << **sizeof**(insulation_tester) << endl; **return** 0; }.
Insulated.h: #ifndef INSULATED_H_ #define INSULATED_H_ **template**< int > class ...
c2.com/cgi/wiki?ExtremeCppFormTest - 7k - [Cached](#) - [Similar pages](#)

[\[PDF\] Microsoft PowerPoint - modernC++ SS04](#)

File Format: PDF/Adobe Acrobat - View as HTML
... Policies and Policy Classes **template** <class T> struct MallocCreator { static T*
Create() { void* buf = std::malloc(**sizeof**(T)); if (!buf) **return** 0; **return** new ...
www.informatik.hu-berlin.de/sam/lehre/mod-c++/modernC++_SS04.5.pdf - [Similar pages](#)

[CGAL Basic Library Manuals:](#)

... Operations. **template** <class SpatialSeparator>. void, ... Point_list::iterator. c.begin (),
Return the iterator pointing to the pointer to the first point. ...
www.cgal.org/Manual/doc_html/basic_lib/Spatial_searching_ref/Class_Point_container.html - 20k - [Cached](#) - [Similar pages](#)

[House Design Templates, Furniture Templates, Landscape Templates ...](#)

... Satisfaction Guaranteed If you are not delighted with our **template** kits, then **return**
the product to us within 30 days for a refund (excluding shipping cost). ...
www.homeplanfinder.com/templates/default.asp - 21k - Sep 17, 2004 - [Cached](#) - [Similar pages](#)

[enum_iterator.h: IEnumXxx container to produce STL-compatible ...](#)

... if _MSC_VER>1020 **template**<> #endif class ... if (p1->lpszVerbName == NULL) **return**;
p1->lpszVerbName = (LPOLESTR)CoTaskMemAlloc(**sizeof**(OLECHAR)*(ocslen ...
www.sellbrothers.com/tools/enum_iterator.h - 16k - [Cached](#) - [Similar pages](#)

[Copyright \(c\) 1994 * Hewlett-Packard Company * * Permission to use ...](#)

... 0); T* tmp = (T*)(::operator new((unsigned int)(size * **sizeof**(T)))); if (tmp ==
0) { cerr << "out of memory" << endl; exit(1); } **return** tmp; } **template** <class T ...
www.cs.rpi.edu/~wiseb/stl/stl/defalloc.h - 5k - [Cached](#) - [Similar pages](#)

[Example: Converting a regular class into a **template**. \(Array\)](#)

... are not equal **return** true; // arrays are equal } // Overloaded subscript operator
for non-const Arrays // reference **return** creates an lvalue **template** <class T ...
www.cs.odu.edu/~wild/cs333/Fall03/cgi-bin/visitweb.cgi?html=content/examples/arraytemplate.htm - 16k -
[Cached](#) - [Similar pages](#)

template return sizeof

Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google



Subscribe Register Login
(Full Service) (Limited Service, Free)

Search: ☒ The ACM Digital Library ☐ The Guide

Optimizing Dynamically <and> Runtime Type Feedback

THE ACM DIGITAL LIBRARY

Feedback

Terms used Optimizing Dynamically and Runtime Type Feedback

Sort results
by

relevance

☒ Save results to a Binder

Try

☒ Search Tips

Try

☐ Open results in a new window

Display results

expanded form

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7 8 9

Best 200 shown

1 Kaleidoscope: mixing objects, constraints, and imperative programming

Bjorn N. Freeman-Benson

September 1990 ACM SIGPLAN Notices , Proceedings of the European conference on
Object-oriented programming systems, languages, and applications

Full text available: pdf(1.14 MB)

Additional Information: full citation, abstract, references, ci

Kaleidoscope is an object-oriented language being designed to integrate the ti
paradigm with the less traditional declarative constraint paradigm. Imperative
declarative constraints provide object relations. A variables as streams seman
integration. A running example is used to illustrate the language concepts&mc

2 1999: A retrospective on: "an evaluation of staged run-time optimizations in

Brian Grant, Matthai Philipose, Markus Mock, Craig Chambers, Susan J. Eggers
April 2004 ACM SIGPLAN Notices, Volume 39 Issue 4

Full text available: pdf(2.18 MB)

Additional Information: full citation, abstract, re

Previous selective dynamic compilation systems have demonstrated that dyna
performance improvements at low cost on small kernels, but they have had di
overcome this limitation, we developed DyC, a selective dynamic compilation
and flexible analyses and transformations. DyC is able to achieve good perform
are much larger and more complex than the kernels. We ...

3 1989: A retrospective on: "customization: optimizing compiler technology for object-oriented programming language"

Craig Chambers, David Ungar

April 2004

ACM SIGPLAN Notices, Volume 39 Issue 4

Full text available:  pdf(2.52 MB)

Additional Information: full citation, abstract, re

Dynamically-typed object-oriented languages please programmers, but their performance. Our new implementation techniques extract static type information from the system compiles several copies of a given procedure, each *customized* for one receiver is bound at compile time. The compiler *predicts* types that are static at run-time type tests to v ...

4 Strength reduction for loop-invariant types

Phung Hua Nguyen, Jingling Xue

January 2004

Proceedings of the 27th conference on Australasian computer science

Full text available:  pdf(147.42 KB)

Additional Information: full citation, abstract, re

Types are fundamental for enforcing levels of abstraction in modern high-level lower-level representations. However, some type-related features such as dynamic casts can contribute substantially to the performance of a program. Loop-invariant object whose dynamic type never changes inside a loop. In this case, operations are redundant in the loop. As these operations often ...

Keywords: PRE, inlining, loop-invariant type, strength reduction, type checking

5 Type feedback vs. concrete type inference: a comparison of optimization techniques for object-oriented languages

Ole Agesen, Urs Hölzle

October 1995 ACM SIGPLAN Notices , Proceedings of the tenth annual conference on programming language systems, languages, and applications, Volume 30 Issue 10

Full text available:  pdf(2.27 MB)

Additional Information: full citation, abstract, references, ci

Two promising optimization techniques for object-oriented languages are type prediction) and concrete type inference (static analysis). We directly compare effectiveness on a suite of 23 SELF programs while keeping other factors constant. We inline over 95% of all sends and deliver similar overall performance with one machine integer ...

6 Selective specialization for object-oriented languages

Jeffrey Dean, Craig Chambers, David Grove

June 1995 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 conference and implementation, Volume 30 Issue 6

Full text available:  pdf(1.32 MB)

Additional Information: full citation, abstract, references, ci

Dynamic dispatching is a major source of run-time overhead in object-oriented languages. It is due to the indirect effect of preventing other optimizations. Compilers for object-oriented languages analyze the classes of objects stored in memory, bounding the possible classes of message receivers enough so that the compiler can generate a message send at compile time ...

7 Continuous program optimization: A case study

Thomas Kistler, Michael Franz

July 2003 ACM Transactions on Programming Languages and Systems (TOPLAS)

Full text available:  pdf(877.67 KB)

Additional Information: full citation, abstract, references, ci

Much of the software in everyday operation is not making optimal use of the hardware. Among the reasons for this discrepancy are hardware/software mismatches, non-optimal software engineering considerations, and the inability of systems to adapt to changing problems is to delay code generation until load time. This is the earliest point at which the system is fine-tuned to the actual capabilities of the hardware ...

Keywords: Dynamic code generation, continuous program optimization, dynamic

8 Practicing JUDO: Java under dynamic optimizations

Micha? Cierniak, Guei-Yuan Lueh, James M. Stichnoth

May 2000 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference and implementation, Volume 35 Issue 5

Full text available:  pdf(190.06 KB)


Additional Information: full citation, abstract, references, ci

A high-performance implementation of a Java Virtual Machine (JVM) consists of several components: Just-In-Time (JIT) compilation, exception handling, synchronization mechanisms, and garbage collection. These components are tightly coupled to achieve high performance. In this paper, we describe techniques implemented in the JIT compilation and exception handling of the Machine (MRL VM), ...

9 Efficient multiple and predicated dispatching

Craig Chambers, Weimin Chen

October 1999 ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on programming languages, systems, languages, and applications, Volume 34 Issue 10

Full text available:  pdf(2.41 MB)


Additional Information: full citation, abstract, references, citations, etc.

The speed of message dispatching is an important issue in the overall performance of object-oriented systems. We have developed an algorithm for constructing efficient dispatch functions that support single dispatching, multiple dispatching, and predicate dispatching. Our algorithm is based on a general predicate dispatching model (which generalizes single dispatching, multiple dispatching, and pattern matching ...)

10 Vortex: an optimizing compiler for object-oriented languages

Jeffrey Dean, Greg DeFouw, David Grove, Vassily Litvinov, Craig Chambers

October 1996 ACM SIGPLAN Notices , Proceedings of the 11th ACM SIGPLAN conference on programming languages, systems, languages, and applications, Volume 31 Issue 10

Full text available:  pdf(2.45 MB)

Additional Information: full citation, abstract, references, citations, etc.

Previously, techniques such as class hierarchy analysis and profile-guided recompilation have been demonstrated to greatly improve the performance of applications written in pure object-oriented languages. To the degree to which these results are transferable to applications written in hybrid languages, we have developed the Vortex compiler infrastructure, a compiler for object-oriented languages, with ...

11 Optimizing dynamically-dispatched calls with run-time type feedback

Urs Hölzle, David Ungar

June 1994 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1994 conference on programming languages and implementation, Volume 29 Issue 6


Full text available:  pdf(1.39 MB)

Additional Information: full citation, references, citations, etc.

12 Debugging optimized code with dynamic deoptimization

Urs Hölzle, Craig Chambers, David Ungar

July 1992 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1992 conference on programming languages and implementation, Volume 27 Issue 7

Full text available:  pdf(1.26 MB)


Additional Information: full citation, abstract, references, citations, etc.

SELF's debugging system provides complete source-level debugging (expected to be available in the near future). It shields the debugger from optimizations performed by the compiler by dynamically deoptimizing code. Deoptimization only affects the procedure activations that are actively being debugged. Deoptimization requires the compiler to supply debugging information

13 A comparison of algorithms for interprocedural class analysis

David Grove

November 1996 Proceedings of the 1996 conference of the Centre for Advanced S

Full text available:  pdf(56.87 KB)



Additional Information: full citation, abstract, reference

Message passing overhead is often a substantial source of runtime overhead in combat this performance problem, a number of techniques have been developed statically-bound procedure calls, which are then amenable to traditional compilation expansion. In this paper, we examine one such technique, interprocedural class analysis for describing interprocedural class analysis algorithms ...

14 Rapid profiling via stratified sampling

S. Subramanya Sastry, Rastislav Bodík, James E. Smith

May 2001 ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual Computer architecture, Volume 29 Issue 2

Full text available:  pdf(1.02 MB)  Publisher Site

Additional Information: full citation, abstract, reference

Sophisticated binary translators and dynamic optimizers demand low overhead, high accuracy, and the ability to collect a variety of performance data that achieves these goals is proposed. Conceptually, the hardware profile data by counting identical events; the compressed profile analysis. Compressing the high-bandwidth event stream greatly. Because optimizations can tolerate ...

15 Whole program compilation for embedded software: the ADSL experiment

A. Johan Cockx

April 2001 Proceedings of the ninth international symposium on Hardware/software

Full text available:  pdf(353.80 KB)

Additional Information: full citation, abstract, reference

The increasing complexity and decreasing time-to-market of embedded software designers to write more modular and reusable code, using for example techniques and languages such as C++. The resulting memory overhead removed by traditional optimizing compilers; a global, whole program analysis to evaluate the potential of whole program optimization technique applied to the embedded software of a commercial ADSL modem. Using ...

Keywords: C++, embedded software, interprocedural optimization

16 A study of devirtualization techniques for a Java Just-In-Time compiler

Kazuaki Ishizaki, Motohiro Kawahito, Toshiaki Yasue, Hideaki Komatsu, Toshio Nakatani
October 2000 ACM SIGPLAN Notices , Proceedings of the 15th ACM SIGPLAN conference on programming
systems, languages, and applications, Volume 35 Issue 10

Full text available:  pdf(225.89 KB)

Additional Information: full citation, abstract, references,

Many devirtualization techniques have been proposed to reduce the runtime overhead of
various object-oriented languages, however, most of them are less effective in a
straightforward manner. This is partly because Java is a statically-typed language and
a call to a static one does not make a tangible performance gain (owing to the lookup in the
table) unless it is inlined, and partly because the JVM is not designed to be fast ...

17 Efficient message dispatch in object-oriented systems

Mayur Naik, Rajeev Kumar

March 2000

ACM SIGPLAN Notices, Volume 35 Issue 3

Full text available:  pdf(906.47 KB)

Additional Information: full citation, abstract, citing

Single dispatch involves performing at run-time a multi-way switch over the parameters of
object-oriented systems implement this switch as an array lookup using a table, or
using a tree-based technique. However, each of these is the best choice only in some
outperforms the other under all circumstances. In this paper, we present a technique for
the switch that employs the table ...


Keywords: implementation, message dispatch, multiple dispatch, object-oriented

18 Design, implementation, and evaluation of optimizations in a just-in-time compiler

Kazuaki Ishizaki, Motohiro Kawahito, Toshiaki Yasue, Mikio Takeuchi, Takeshi Onodera,
Hideaki Komatsu, Toshio Nakatani

June 1999

Proceedings of the ACM 1999 conference on Java Grande


Full text available:  pdf(1.09 MB)

Additional Information: full citation, references, citations, i

19 Efficient dynamic dispatch without virtual function tables: the SmallEiffel compiler

Olivier Zendra, Dominique Colnet, Suzanne Collin

October 1997 ACM SIGPLAN Notices , Proceedings of the 12th ACM SIGPLAN conference on
systems, languages, and applications, Volume 32 Issue 10

Full text available:  pdf(2.10 MB)


Additional Information: full citation, abstract, references, ci

SmallEiffel is an Eiffel compiler which uses a fast simple type inference mechanism
replacing them by static bindings. Starting from the system's entry point, it computes
saves compiling and then removing dead code. As the whole system is analyzed
and genericity do not cause any overhead. SmallEiffel features a coding scheme
function tables. Dynamic dispatch ...

20 The direct cost of virtual function calls in C++

Karel Driesen, Urs Hölzle

October 1996 ACM SIGPLAN Notices , Proceedings of the 11th ACM SIGPLAN conference on programming languages, systems, languages, and applications, Volume 31 Issue 10

Full text available:  pdf (2.03 MB)

Additional Information: full citation, abstract, references, citation metrics




We study the direct cost of virtual function calls in C++ programs, assuming that virtual function tables are not shared. We measure this overhead experimentally for a number of programs and a combination of executable inspection and processor simulation. Our results show that on average, virtual function calls spend a median of 5.2% of their time and 3.7% of their instructions in dispatch. In some programs, the median overhead rises to 13. ...

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7

The ACM Portal is published by the Association for Computing Machinery. C

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows Media



Subscribe Register
(Full Service) (Limited Service, Free)

Login

Search: ☒ The ACM Digital Library ☐ The Guide

size operator at runtime

THE ACM DIGITAL LIBRARY

Feedback

Terms used size operator at runtime

Sort results
by

relevance

☒ Save results to a Binder

Try

☒ Search Tips

Try

☐ Open results in a new window

Display results

expanded form

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7 8 9

Best 200 shown

1 Stream query processing II: Chain: operator scheduling for memory minimi

Brian Babcock, Shivnath Babu, Rajeev Motwani, Mayur Datar

June 2003 Proceedings of the 2003 ACM SIGMOD international conference on M

Full text available: pdf(299.62 KB)

Additional Information: full citation, abstract, references, c

In many applications involving continuous data streams, data arrival is bursty. Systems that seek to give rapid or real-time query responses in such an environment gracefully with bursts in data arrival without compromising system performance processing bursty streams --- *adaptive, load-aware scheduling* of query operators during times of peak load. We show that the cho ...

2 Quality of service in an information economy

R. Braumandl, A. Kemper, D. Kossmann

November 2003 ACM Transactions on Internet Technology (TOIT), Volume 3

Full text available: pdf(829.15 KB)

Additional Information: full citation, abstract, references, c

Accessing and processing distributed data sources have become important factors, especially true for the emerging virtual enterprises with their data and processes on the Internet. Unfortunately, however, query processing on the Internet is not precise enough to meet the requirements of many business applications. For instance, the response time of a query and the monetary cost might be too high if the ...

Keywords: Quality of Service

3 Compilation of optimized OBDD-algorithms

S. Höreth

September 1996 Proceedings of the conference on European design automation

Full text available:  pdf(397.77 KB) Additional Information: full citation, references, index terms

4 Vector reduction/transformation operators

Roscoe A. Bartlett, Bart G. Van Bloemen Waanders, Michael A. Heroux

March 2004 ACM Transactions on Mathematical Software (TOMS), Volume 30

Full text available:  pdf(516.85 KB) Additional Information: full citation, abstract, reference


Development of flexible linear algebra interfaces is an increasingly critical issue. Interfaces that are well established for some linear algebra abstractions, but not for vectors. Due to the diversity of necessary operations, sometimes requiring dozens for a given abstraction (for optimization). We discuss a new approach based on operator objects that are supported by the linear algebra lib ...

Keywords: Optimization, interfaces, object-orientation, vectors

5 Fast and Efficient Construction of BDDs by Reordering Based Synthesis

A. Hett, R. Drechsler, B. Becker

March 1997 Proceedings of the 1997 European conference on Design and Test

Full text available:  pdf(913.25 KB)  Publisher Site Additional Information: full citation, abstract, reference

We present a new approach to symbolic simulation with BDDs. Our method uses a reordering algorithm which allows the integration of dynamic variable ordering (even within a single AND-operation). Thus, huge peak sizes during the construction can often be avoided. This approach, with no penalty in runtime, is more memory efficient than traditional ITE operations. Results are confirmed by experiments on a large set of ...

Keywords: data structures, BDD, reordering based synthesis, symbolic simulation, binary decision diagram

6 Asserting performance expectations

Jeffrey S. Vetter, Patrick H. Worley

November 2002 Proceedings of the 2002 ACM/IEEE conference on Supercomputing



Full text available:  pdf(351.59 KB) Additional Information: full citation, abstract, reference

Traditional techniques for performance analysis provide a means for extracting performance information from applications. Users then compare this raw data to their performance constructs. This comparison can be tedious for the scale of today's architectures. In this situation, we present a methodology and prototype that allows users to assert performance expectations in their source code using performance assertions ...

7 GlueQoS: Middleware to Sweeten Quality-of-Service Policy Interactions

May 2004

Proceedings of the 26th International Conference on Software Engin

Full text available:  pdf(652.24 KB)  Publisher Site

Additional Information:

A holy grail of component-based software engineering is "write-once, reuse everywhere": distributed, component-based systems supporting emerging application areas: (where web services are viewed as components) and Peer-to-Peer computing, requirements (related to quality-of-service (QoS) issues such as security, reliability, deployment context, and sometimes even at run-time, complicating t ...

8 Research sessions: non-standard query processing: Buffering database operations and cache performance

Jingren Zhou, Kenneth A. Ross

June 2004

Proceedings of the 2004 ACM SIGMOD international conference on M

Full text available:  pdf(188.52 KB)

Additional Information: full citation, abstract, n

As more and more query processing work can be done in main memory access, the cache component of database operations. Recent database research has shown that second-level cache data misses and first-level instruction cache misses. While reducing the data cache misses, relatively little research has been done on improving the performance of database systems. We first answer the question "Why ...

9 Streams: PIPES: a public infrastructure for processing and exploring stream data

Jürgen Krämer, Bernhard Seeger

June 2004

Proceedings of the 2004 ACM SIGMOD international conference on M

Full text available:  pdf(336.66 KB)

Additional Information: full citation, abstract,

PIPES is a flexible and extensible infrastructure providing fundamental building blocks for a data management system (DSMS). It is seamlessly integrated into the Java library for stream processing and extends XXL's scope towards continuous data-driven query processing.

10 Techniques for the translation of MATLAB programs into Fortran 90

Luiz De Rose, David Padua

March 1999

ACM Transactions on Programming Languages and Systems (TOPLAS)

Full text available:  pdf(467.60 KB)

Additional Information: full citation, abstract, references, cit

This article describes the main techniques developed for FALCON's MATLAB-to-Fortran 90 translation. It provides a programming environment for the development of high-performance scientific applications. It uses dynamic inference methods to translate MATLAB programs into Fortran 90. The article also describes advanced value propagation techniques and symbolic algorithms for subscript reduction. FALCON's MATLAB translator can generate code that performs much faster than the original MATLAB code.

Keywords: MATLAB, array language compilation, inference

11 A configurable type hierarchy index for OODB

Thomas A. Mueck, Martin L. Polaschek

November 1997 The VLDB Journal — The International Journal on Very Lar

Full text available:  pdf(411.47 KB)

Additional Information: full citation, abstract, citing

With respect to the specific requirements of advanced OODB applications, index OODBMS have to provide efficient support for multiattribute queries and have particular query profile. We describe the *multikey type index* and an efficient i It meets both requirements: in addition to its multiattribute query capabilities two standard design altern ...

Keywords: Access methods, Indexing, Multiple inheritance, OODB, Type hierar

12 Efficient mid-query re-optimization of sub-optimal query execution plans

Navin Kabra, David J. DeWitt

June 1998 ACM SIGMOD Record , Proceedings of the 1998 ACM SIGMOD internati data, Volume 27 Issue 2

Full text available:  pdf(1.83 MB)

Additional Information: full citation, abstract, references, ci

For a number of reasons, even the best query optimizers can very often produ leading to a significant degradation of performance. This is especially true in c support queries and/or object-relational databases. In this paper, we describe sub-optimality of a query execution plan during query execution and attempts is to collect statistics at key points durin ...

13 Orchestrating interactions among parallel computations

Susan L. Graham, Steven Lucco, Oliver Sharp

June 1993 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 confere and implementation, Volume 28 Issue 6

Full text available:  pdf(1.12 MB)

Additional Information: full citation, abstract, references, ci

Many parallel programs contain multiple sub-computations, each with distinct requirements. The traditional approach to compiling such programs is to impo between sub-computations, optimizing each as a separate entity. This paper c the interactions among sub-computations, avoiding strict synchronization whe are possible. Our approach to c ...

- 14 The design and implementation of a parallel array operator for the arbitrary**
Steven J. Deitz, Bradford L. Chamberlain, Sung-Eun Choi, Lawrence Snyder
June 2003 ACM SIGPLAN Notices , Proceedings of the ninth ACM SIGPLAN sympos
parallel programming, Volume 38 Issue 10

Full text available:  pdf(244.89 KB)

Additional Information: full citation, abstract, referen

Gather and scatter are data redistribution functions of long-standing importan
this paper, we present a highly-general array operator with powerful gather an
other array languages. We discuss an efficient parallel implementation, introd
optimizations---schedule compression, dead array reuse, and direct communic
with the operator's wide applicability. In our implementation ...

Keywords: ZPL, array languages, gather, parallel programming, scatter

- 15 Memory characteristics of iterative methods**

Christian Weiß, Wolfgang Karl, Markus Kowarschik, Ulrich Rude

January 1999 Proceedings of the 1999 ACM/IEEE conference on Supercomputing (

Full text available:  pdf(438.45 KB)

Additional Information: full citation, references, citings, index terms

- 16 High level specification and design: High-Level specification and automatic**

Marcio T. Oliveira, Alan J. Hu

June 2002 Proceedings of the 39th conference on Design automation

Full text available:  pdf(96.42 KB)

Additional Information: full citation, abstract, reference

A central problem in functional verification is to check that a circuit block is pr
that the environment is providing legal inputs. To attack this problem, several
monitor-based methodologies, which offer many benefits. This paper presents
for these monitors, along with a linear-size, linear-time translation algorithm
style naturally fits the complex, ...

Keywords: alternation, formal verification, pipelining, regular expressions

- 17 The 1999 ICFP programming contest**

Norman Ramsey, Kevin Scott


March 2000 ACM SIGPLAN Notices, Volume 35 Issue 3

Full text available:  pdf(1.10 MB) Additional Information: full citation, index terms

18 Processing aggregate relational queries with hard time constraints

Wen-Chi Hou, Gultekin Ozsoyoglu, Baldeo K. Taneja

June 1989 ACM SIGMOD Record , Proceedings of the 1989 ACM SIGMOD international conference on database management systems, Volume 18 Issue 2

Full text available:  pdf(1.26 MB)

Additional Information: full citation, abstract, references, citation

We consider those database environments in which queries have strict timing constraints. We develop a time-constrained query evaluation methodology. For aggregate relational algebra queries, we develop a time-constrained query evaluation algorithm. The algorithm, which is implemented in a prototype system, samples from input relations, and evaluates the associated estimators developed. The algorithm uses a stopping criterion (e.g., a time quota or a desired error range ...

19 Polymorphic splitting: an effective polyvariant flow analysis

Andrew K. Wright, Suresh Jagannathan

January 1998 ACM Transactions on Programming Languages and Systems (TOPLAS)

Full text available:  pdf(517.76 KB)

Additional Information: full citation, abstract, references, citation


This article describes a general-purpose program analysis that computes global flow information for higher-order, call-by-value languages. The analysis employs a novel form of polymorphic splitting that uses let-expressions as syntactic clues to gain precision. The information is used both to eliminate run-time checks and to inline procedure calls. The analysis is implemented in a suite of Scheme programs ...

Keywords: flow analysis, inlining, polyvariance, run-time checks

20 A call to order

David Maier, Bennet Vance

August 1993 Proceedings of the twelfth ACM SIGACT-SIGMOD-SIGART symposium

Full text available:  pdf(1.49 MB)




Additional Information: full citation, abstract, references, citation

Scientific applications are infrequent users of commercial database management systems because they do not offer good support for ordered data structures, such as multidimensional arrays. A natural representation of many scientific data types. In this paper, we lay out a framework for ordered data structures, consider possible approaches along with their advantages and shortcomings, and survey a wide variety of prior work out ...

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7

The ACM Portal is published by the Association for Computing Machinery. C

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  Adobe Acrobat  QuickTime  Windows Media

[Yahoo!](#) [My Yahoo!](#) [Mail](#) [Welcome, Guest](#) ([Sign In](#))

[Search Home](#) [Help](#)
[Web](#) [Images](#) [Directory](#) [Yellow Pages](#) [News](#) [Products](#)


[Shortcuts](#) [Advanced Search](#) [Preferences](#)

Search Results

Results 1 - 10 of about 10,300 for RTTI operator - 0.35 sec. ([About this page](#))1. [File Member Index](#)

... **operator** &&() : Predicate.h. **operator** delete() : gblnew.h. **operator** delete[]() : gblnew.h. **operator** new ... UTL_Cl
libutl.sourceforge.net/globals.html - 10k - [Cached](#) - [More pages from this site](#)

2. [http://doc.trolltech.com/3.0/index](#)

... QAsciiCacheIterator::**operator**()" qasciiCacheIterator.html#**operator**() "QAsciiCacheIterator::operator++ ... QCanv
doc.trolltech.com/3.0/index - 417k - [Cached](#) - [More pages from this site](#)

3. [C++ User's Guide: 5 - Runtime Type Information](#)

Runtime Type Information. 5. This chapter explains the use of Runtime Type Information (RTTI). ... RTTI Options. R1
www.ictp.trieste.it/~manuals/programming/sun/c-plusplus/c++_ug/RTTI.doc.html - [More pages from this site](#)

4. [OS-FAQ Wiki - Doing a kernel in C++](#)

... implement some memory management, and the **operator** new() and **operator** delete() functions (including their ar
www.mega-loskyo.com/osfaq2/index.php/Doing%20a%20kernel%20in%20C++ - 21k - [Cached](#) - [More pages from this](#)

5. [8: Run-time type identification](#)

... Run-time type identification (RTTI) lets you find the exact type of an object when you have only ... The typeid() op
www.cs.umd.edu/class/spring2000/cmsc433/TIC/V2/Chapter08.html - 70k - [Cached](#) - [More pages from this site](#)

6. [qcanvas.h Include File](#)

This is the verbatim text of the qcanvas.h include file. It is provided only for illustration; the copyright remains with Trc
doc.trolltech.com/qcanvas-h.html - 23k - [Cached](#) - [More pages from this site](#)

7. [9: Run-time type identification](#)

Thinking in C++, 2nd edition, Volume 2. Revision 4.0. by Bruce Eckel & Chuck Allison. ©2001 MindView, Inc. 9: Run-
you have only ... this way it works just like the C++ dynamic_cast **operator**. [[Comment](#) ...
www.belayenterprises.com/ticpp_v2/Chap09.htm - 87k - [Cached](#) - [More pages from this site](#)

8. [Self-Similar](#)

Abstract: we describe how C++ **operator** dynamic_cast<> can be implemented. We also present "**operator**" guid_ca
called RTTI for RunTime Type Information). Whenever a class has at ...
www.self-similar.com/dynamic_cast.html - 11k - [Cached](#) - [More pages from this site](#)

9. [FieldPosition class Reference](#)

... **operator**= (const FieldPosition &) Assignment **operator**. More... ... ICU "poor man's RTTI", returns a UClassID for
oss.software.ibm.com/icu/api/previous/classFieldPosition.html - 36k - [Cached](#) - [More pages from this site](#)

10. [Your.exe - 0 error\(s\), 0 warning\(s\)](#)

C++ Tips and tricks, for experts and beginners. ... pointer to a base class, you can use RTTI to determine exactly whi
www.buginthemachine.com/progcpp2.php - 17k - [Cached](#) - [More pages from this site](#)

Results Page:

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [Next](#)
Tip: You can change the number of results on each page in [preferences](#).
[Web](#) [Images](#) [Directory](#) [Yellow Pages](#) [News](#) [Products](#)

Your Search: RTTI operator

Yahoo! My Yahoo! Mail Welcome, Guest [Sign In]

Search Home Help

Web Images Directory Yellow Pages News Products











YAHOO! search runtime sizeof operator

Search

Shortcuts Advanced Search Preferences

Search Results

Results 11 - 20 of about 11,800 for runtime sizeof operator - 0.07 sec. (About this page)

11. [Comp.compilers: Re: Help: How to determine big/little endian?](#) 
... Even, the **sizeof operator** isn't strictly necessary: you could easily, determine the **sizeof** a type at **runtime** with a
iecc.com/comparch/article/95-09-057 - 4k - [Cached](#) - [More pages from this site](#)
12. [3](#) 
... tools that allow memory to be allocated a **runtime**. So-called dynamic memory is memory requested by ... The siz
www.gradel.org/mccc/POINTER.htm - 202k - [Cached](#) - [More pages from this site](#)
13. [Boost Compiler Status Error Log](#) 
Compiler Status: linux. Run Date: 06:24:57 UTC, Saturday 17 July 2004. Compiler output:
boost.sourceforge.net/regression-logs/cs-Linux-links.html - 525k - [Cached](#) - [More pages from this site](#)
14. [Computing Memory Alignment Restrictions of Data Types in C and C++](#) 
... structures of heterogenous data are assembled at **runtime**, one must respect memory alignment restrictions of the
cl.com.com/click?q=df-1k~hQLElxoaAWdlsi_3cJU16h6fz - 10k - [Cached](#) - [More pages from this site](#)
15. [Boost Compiler Status Error Log](#) 
Compiler Status: Generic Unix. Run Date: 20:02:56 UTC, Tuesday 04 February 2003. Run output:
www.comp-phys.org/boost/cs-sn9626-links.html - 528k - [Cached](#) - [More pages from this site](#)
16. <http://web.cs.ualberta.ca/~ernie/201/lab4> 
... which an array variable points CAN NOT be changed at **runtime**. * The memory that a pointer points to ... ptr plus
web.cs.ualberta.ca/~ernie/201/lab4 - 2k - [Cached](#) - [More pages from this site](#)
17. [Boost Compiler Status Error Log](#) 
Compiler Status: Mac OS. Run Date: 18:40:44 UTC, Saturday 21 August 2004. Compiler output:
www.boost.org/regression-logs/cs-Darwin-links.html - 231k - [Cached](#) - [More pages from this site](#)
18. [Boost Compiler Status Error Log](#) 
Compiler Status: Mac OS. config - config_info - darwin. Compiler output: /
boost.sourceforge.net/regression-logs/cs-Darwin-links.html - 231k - [Cached](#) - [More pages from this site](#)
19. <http://www.artsci.wustl.edu/~eawelsh/mem.ppt> (MICROSOFT POWERPOINT) 
... varies from computer to computer. The **sizeof()** operator returns the size, in bytes, of a variable type ... well the a
www.artsci.wustl.edu/~eawelsh/mem.ppt - 50k - [View as html](#) - [More pages from this site](#)
20. <http://www.lcdf.org/c++/clauseE.html> 
(normative)Universal-character-names. 1 This clause lists the complete set of hexadecimal code values that are valid
and logical 88 **sizeof** 77 and pointer to ... ambiguity 167 **runtime_error** 351 template name 257 **runtime_error** 351 to
www.lcdf.org/c++/clauseE.html - 239k - [Cached](#) - [More pages from this site](#)

Results Page:

Prev ◀ 1 2 3 4 5 6 7 8 9 10 ▶ Next

Tip: You can change the number of results on each page in preferences.

Web Images Directory Yellow Pages News Products

Your Search: runtime sizeof operator

Search